

Comparison of Solvers for Contact Problems

Anilraj Sudhakar^{1,a)}

¹*IT4Innovations National Supercomputing Center, VSB - Technical University of Ostrava, 17. listopadu 2172/15,
70800 Ostrava, Czech Republic*

^{a)}Corresponding author: anilraj.sudhakar@vsb.cz

Abstract. The paper focuses on the comparison of sequential solvers used for solving quadratic programming problems which have a wide variety of applications including contact modeling. The complex algebraic formulation generated from the contact problems may be reduced to simple quadratic programming problem using different domain decomposition methods and can be effectively solved using different solvers. Such solvers are applied on benchmark problems and the achieved results are compared.

INTRODUCTION

One of the most thought-provoking tasks associated with the analysis of rotor dynamic systems is the numerical modeling of the contacts. These problems are difficult to formulate and solve because of the non-linearity associated in the relations between displacements or rates of displacements and forces on a part of the boundary which is usually governed by tribological laws. The discretization of such problems leads to very large and ill-conditioned systems. Such systems can be transformed by dualization theory to smaller better conditioned systems by employing domain decomposition methods with asymptotically linear complexity. A powerful tool for the massively parallel solution of the contact problems is the Finite Element Tearing and Interconnecting (FETI) methods. The key ingredient of FETI is the decomposition of the spatial domain into non-overlapping subdomains that are glued by Lagrange multipliers. The original problem, after reducing the primal variables, even reduces to a smaller, relatively well-conditioned equality constrained Quadratic Programming (QP) problem which can be solved using different iterative solvers. The important advantage of applying FETI procedure to the contact problems is that, in addition to duality reducing the dimension of the original problem, it also turns all the inequality constraints into bound constraints. Also, FETI methods provide numerical scalability for the problem. Even though the resulting QP problem has bound constraints along with the equality constraints, the resulting problem is still easier to solve compared to the contact problems in displacements.

Quadratic Programming Problems

The original contact problem, after discretization using domain decomposition methods like FETI or a variant of FETI, known as Total FETI (TFETI) which enforces the Dirichlet boundary conditions with the help of Lagrange multipliers; is reduced to a quadratic programming problem with equality and bound constraints. The QP problem obtained is basically a constrained minimization problem that can be solved using different optimization algorithms. The discretized problem generally consists of a quadratic cost function $f(x)$, an equality constraint and non-negativity constraints. The problem is likely in the following form:

$$\text{minimize } f(x); \text{ subject to } x \in \Omega_{BE} \quad (1)$$

with $f(x) = 1/2 x^T A x - b^T x$, A is a symmetric positive definite matrix, $b \in \mathbb{R}^n$, $\Omega_{BE} = \{x \in \mathbb{R}^n : lb \leq x \leq ub \text{ and } Cx = d\}$ and $C \in \mathbb{R}^{m \times n}$. There are different iterative algorithms available for solving such minimization problem.

Some of the significant algorithms are discussed in this section and a basic insight on the solution of the problems using these algorithms are also provided.

QPCE Algorithm

QPCE is a variant of the Semi-Monotonous Augmented Lagrangians with Bound and Equality constraints (SMALBE). It is specifically designed for quadratic problems with bound and equality constraints and makes use of the augmented Lagrangian method. SMALBE generates approximations of the Lagrange multipliers in the outer loop while the bound constraint auxiliary problems are solved approximately in the inner loop.

The auxiliary problem reads *minimize* $L(x, \lambda^k, \%)$ *subject to* $x \geq l$ where

$$L(x, \lambda^k, \%) = f(x) + (\lambda^k)^T (Bx) + \frac{\%}{2} Bx^T Bx. \quad (2)$$

The SMALBE algorithm generates approximations of Lagrange multipliers in the outer loop by means of approximate solutions of the auxiliary problems. The algorithm starts with an inner iteration with adaptive precision control and finding the x value for the auxiliary problem satisfying the condition that the projected gradient is less than or equal to the minimum value of the problem. Effective preconditioners are employed to improve the performance of the minimization. Convergent algorithms are used for minimizing the strictly convex quadratic function subjected to bound constraints. The process is iterated by modifying the Lagrange multipliers until the stopping criteria are met.

Semi-smooth Newton Method

The Newton method is an iterative method for finding the roots of a differentiable function having solutions to the equation $f(x) = 0$. The algorithm works on the assumption of having a function which is continuously differentiable in the region. However, problems with different types of constraints tend to be non-smooth problems which are not continuously differentiable. This leads to the development of a generalized Newton method, also known as the semismooth Newton Method. It employs a variation of the Newton method that is based on an iterative scheme. The main aim of this method is to introduce several objects from the non-smooth analysis which provide generalizations of the classical differentiability concept. A generalized gradient is introduced which changes in the interval where the function is not continuously differentiable, otherwise will have the same effect of that of the Jacobian in a smooth Newton method.

The basic idea is to reformulate the contact constraints such that a Newton-type algorithm can be applied not only for geometrical or material nonlinearity, but also for nonlinearity stemming from contact. A complementary function is defined for the bound constraints

$$S(\lambda_i, r - |x - c|) = \lambda_i - \max(0, \lambda_i - \alpha(r - |x - c|)), \quad (3)$$

$$\text{where } |x - c| \leq r; c = 1/2(ub + lb); r = 1/2(ub - lb). \quad (4)$$

This reduces the system to a minimization problem *minimize* $L(x, \lambda_E, \lambda_I)$ *subject to* $\lambda_I \geq 0$. The generalized Jacobian is formed from the given conditions and solved with necessary stopping criteria.

Matlab Quadratic Programming Method

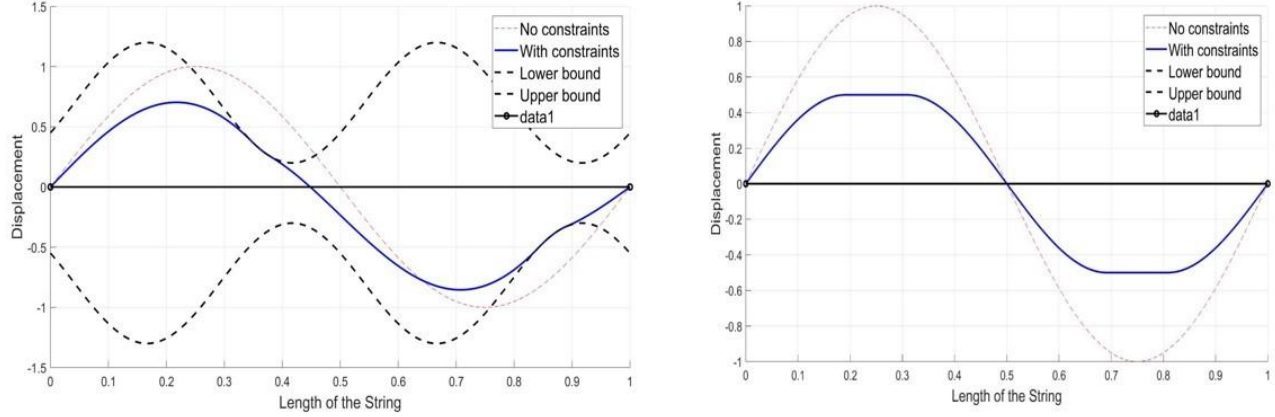
Quadratic Programming method, or briefly quadprog, is a built-in function of Matlab. The algorithm is used for solving only quadratic objective functions with linear constraints. The algorithm finds a minimum for a problem specified by

$$\min_x \frac{1}{2} x^T H x + f^T x \text{ such that } A \cdot x \leq b; A_{eq} \cdot x = b_{eq}; lb \leq x \leq ub. \quad (5)$$

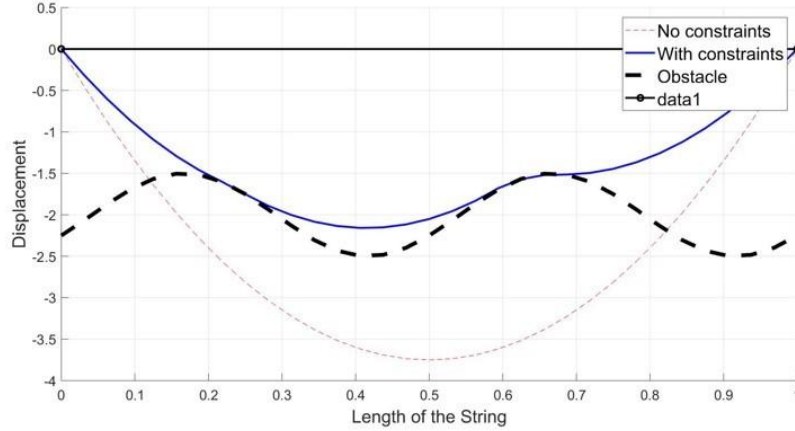
Quadprog applies only to the solver based optimization problem setup. This setup represents the objective and constraints as functions or matrices and allows the use of Hessian multiply function or Jacobian multiply function to save memory of large problems. The algorithm resets components of the initial value x_0 that violates the bounds $lb \leq x \leq ub$ to the interior of the box defined by the bounds.

Numerical Problems

Different methods explained in the previous section are used to solve different benchmark problems and compared to understand the difference in the performance of the algorithms. The first problem is a basic one-dimensional string system with unit length having box constraints and harmonic force. The problem is described by a partial differential equation of the second-order with box constraints on the displacements. The algebraic formulation can be converted into a quadratic programming problem enabling it to be solved using the proposed algorithms. Necessary constraints will be enforced by Lagrange multipliers. It can be observed that the resulting deflection vary with the type of constraints used in Fig.1. Also, the results obtained using different algorithms are in tandem. The computational time is not showing many variations as the problem is simple without many computational efforts.



Varying Force with Constraints

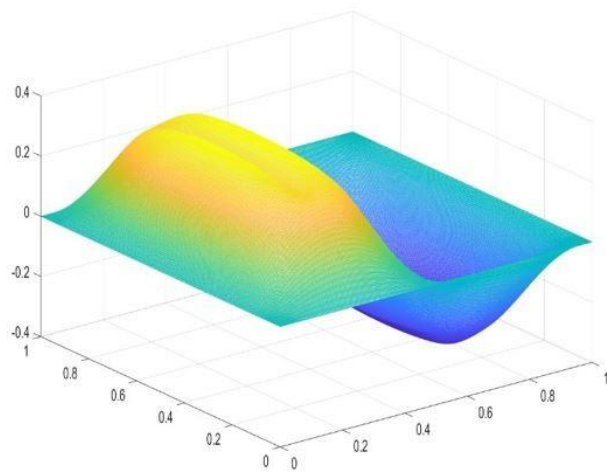


Constant Force with Constraints

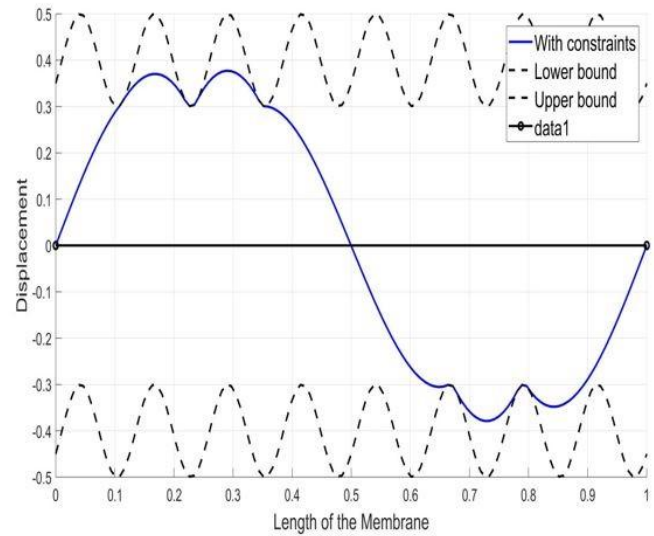
FIGURE 1. Behaviour of string for different types of forces and constraints

The second benchmark problem comprises a two-dimensional membrane having forces acting on the membrane and lower and upper obstacles which play the role of the box constraints. The problem of the membrane is more complex compared to the one-dimensional string system owing to the increase in the degrees

of freedom of the system. However, the problem can be transformed again into a quadratic programming problem with equality and bound constraints similar to the case of a string system (Fig.2).



3 dimensional surface plot



Cross Sectional Plot

FIGURE 2. Response behaviour of two dimensional membrane

Behaviour of the solution time and the number of iterations dependent on the problem size is depicted in Fig 3. As the number of elements increases and the problem becomes more complex, the variation in the solving capacity of different solvers becomes clear and distinct. It can be observed that Semi-smooth newton method which is employed here works faster on the coarser meshes but as the number of elements increases, the performance of these solvers tends to get affected, unlike the other iterative solvers which show much better performance for finer meshes. Let us also mention that the results are affected by the non-optimal implementation in Matlab.

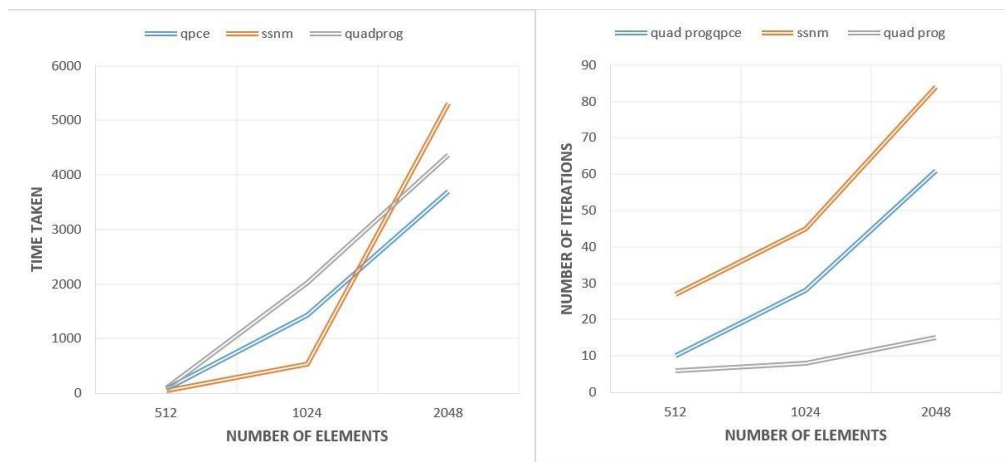


FIGURE 3. Behaviour of Solution time and Iterations for different discretizations

CONCLUSION

Comparison of solvers used for solving problems with contact was essential in understanding the working of different types of algorithms. It gave an idea regarding ways to approach and effectively solve different kinds of contact problems depending on the type of problems and their constraints. Also, the knowledge of the solution times and numbers of iterations and their costs for different algorithms gave us the realization of employing methods like domain decomposition and parallel programming techniques which can guide us in considerably reducing the time and cost of solving huge problems with billions of unknowns in a very effective manner.

The future aim of the research includes understanding and effective implementation of domain decomposition methods like FETI or TFETI in realistic benchmark problems which resembles contact problems with friction and transient contact problems. This can be further improved by employing parallel and efficient solver libraries for dynamic contact problems with friction based on domain decomposition methods which can provide better scalability to the problems. Also, employing these methods in real-life rotor models presents a new challenge. Parallel programming and domain decomposition methods can be effectively combined with solution methods like Harmonic Balance Methods to solve complex contact conditions occurring in rotors. This can be combined with rotor systems already incorporated with non-linearities arising from support systems of shafts, cyclic contacts in the shroud and so on.

ACKNOWLEDGMENTS

This work is a part of the project EXPERTISE that received funding from the European Union H2020 research and innovation program under the MarieSkłodowska-Curie grant agreement No 721865.

REFERENCES

- [1] D. Horak, V. Hapla, J. Kruzik, R. Sojka, M. Cermak, J. Tomcala, M. Pecha, and Z. Dostal, *Advances in Electrical and Electronic Engineering* 15, 230–236 (2017).
- [2] Z. Dostal, D. Horak, and D. Stefanica, *Quadratic programming and scalable algorithms for variational inequalities*, edited by I. . (online) (Springer, 2006), pp. 62–78.
- [3] Z. Dostal, D. Horak, and P. Vodstrcil, *Computational Optimization and Applications* 58, 87–103 (2014).
- [4] Z. Dostal and P. Beremlijski, *Advances in Electrical and Electronic Engineering* 15, 215–222 (2017).
- [5] M. Hintermuller, Department of Mathematics, Humboldt-University, Berlin (2010).
- [6] Z. Dostal and R. Kucera, *SIAM Journal on Optimization* 20, 2913–2938 (2010).
- [7] Haslinger, J., Kucera, R., and Sassi, T., *Math. Model. Nat. Phenom.* 4, 123–146 (2009).